WHAT IS CLAIMED IS:

1. A processor comprising:

a replay system for determining which instructions have not executed properly and replaying those instructions which have not executed properly;

a memory execution unit coupled to the replay system for executing load and store instructions, the memory execution unit including an invalid store flag that is set for a store instruction if the replay system detects that the store instruction has not executed properly and is cleared if the store instruction has executed properly; and

wherein if an invalid store flag is set for a store instruction, the replay system replays load instructions which are programmatically younger than the invalid store instruction so long as the invalid store flag is set.

2. The processor of claim 1 wherein the memory execution unit comprises a memory ordering buffer for maintaining the ordering of load and store instructions and a bus queue for handling bus requests to an external bus.

3. The processor of claim 1 wherein said replay system comprises a checker for determining whether an instruction has executed properly.

4. A processor comprising:

a replay system for determining which instructions have not executed properly and replaying instructions which have not executed properly;

a memory execution unit coupled to the replay system for executing load and store instructions, the memory execution unit including;

a store buffer including one or more entries for storing information related to store instructions, each entry in the store buffer including an invalid store flag that is set for a store instruction if the replay system detects that the store instruction has not executed properly and is cleared if the store instruction has executed properly; and

an external replay signal that is generated from the memory execution unit to the replay system if an invalid store flag is set for a store instruction in the store buffer, the external replay signal being generated for each load instruction that is programmatically younger than the invalid store instruction to allow the replay system to replay the younger load instructions.

5. The processor of claim 4 wherein the replay system generates an invalid store signal to the memory execution unit if the replay system detects that a store instruction has executed improperly, and the memory execution unit then setting the invalid store flag for the store instruction in response to the invalid store signal.

6. The processor of claim 4 wherein the memory execution unit further comprises a bus queue coupled to the store buffer for issuing and tracking memory requests which are sent to an external bus.

7. The processor of claim 6 wherein if there is a store instruction which did not execute properly, the bus queue inhibits memory requests to an external bus for load instructions programmatically younger than the store instruction until the store instruction executes properly.

8. A processor comprising:

a replay system for determining which instructions have not executed properly and replaying instructions which have not executed properly; and

a memory execution unit coupled to the replay system for executing load and store instructions, the memory execution unit including;

a bus queue for issuing and tracking memory requests to an external bus, the bus queue being notified by the replay system if a store instruction executes improperly, the bus queue inhibiting memory requests to the external bus for load instructions that are programmatically younger than the store instruction that executed improperly until the store instruction executes properly.

9. The processor of claim 8 wherein the replay system generates an invalid store signal to the bus queue in response to detecting that a store instruction has executed improperly.

10. The processor of claim 9 wherein the bus queue includes an inhibit load flag that is set when the bus queue receives the invalid store signal and a sequence number field for storing the sequence number of the store instruction that has executed improperly, the bus queue inhibiting memory requests for younger load instructions if the invalid load flag is set.

11. A method of processing instructions comprising:

detecting that a store instruction has executed improperly;

replaying the store instruction; and

replaying subsequently received load instructions which are programmatically younger than the store instruction until the store instruction has executed properly.

12. A method of processing instructions comprising:

detecting that a store instruction has executed improperly because of invalid data;

replaying the store instruction; and

replaying subsequently received load instructions which are programmatically younger than the store instruction and have an address that matches an address of the store instruction.

13. A method of processing instructions comprising:

detecting that a store instruction has executed improperly;

replaying the store instruction; and

inhibiting memory accesses to an external bus for load instructions that are programmatically younger than the store instruction until the store instruction has executed properly.

14. The method of claim 13 and further comprising the step of replaying subsequently received load instructions which are programmatically younger than the store instruction until the store instruction has executed properly.

15. A method of processing instructions comprising:

executing a store instruction;

detecting that the store instruction has not executed properly;

setting an invalid store flag for the store instruction;

routing the store instruction back to an execution unit for replay;

executing a programmatically younger load instruction;

detecting that the store instruction executed improperly based on the invalid store flag;

routing the load instruction back to the execution unit for replay; and

clearing the invalid store flag only after the store instruction has executed properly.

16. A method of processing instructions comprising:

executing a store instruction at a memory execution unit;

detecting at a replay system that the store instruction has not executed properly;

setting an invalid store flag for the store instruction in a memory execution unit;

replaying the store instruction;

executing a load instruction at the memory execution unit that is younger than the store instruction;

the memory execution unit generating an external replay signal to the replay system based on set invalid store flag; and

the replay system replaying the load instruction in response to the external replay signal from the memory execution unit.

17. The method of claim 16 and further comprising the steps of:

detecting that the store instruction has executed properly at replay;

clearing the invalid store flag for the store instruction in a memory execution unit;

allowing the store instruction to retire;

detecting that the load instruction has executed properly at replay;

detecting that the invalid store flag for older store instructions are clear; and

allowing the load instruction to retire.

18. The method of processing instructions of claim 16 and further comprising the steps of:

inhibiting memory accesses to an external bus for the load instruction until the store instruction has executed properly.

19. A processor comprising:

a replay system for determining which instructions have not executed properly and replaying those instructions which have not executed properly;

a memory execution unit coupled to the replay system for executing load and store instructions, the memory execution unit including an invalid address flag that is set for a store instruction if the replay system detects that the store instruction has not executed properly due to an invalid address;

wherein if an invalid address flag is set for a store instruction, the replay system replays load instructions having an address matching the address of the invalid store instruction and which are programmatically younger than the invalid store instruction.

20. A processor comprising:

a replay system for determining which instructions have not executed properly and replaying those instructions which have not executed properly;

a memory execution unit coupled to the replay system for executing load and store instructions, the memory execution unit including an invalid address flag indicating if a store instruction has not executed properly due to an invalid address;

wherein if an invalid address flag is set for a store instruction, the replay system replays load instructions having an address matching the address of the invalid store instruction and which are programmatically younger than the invalid store instruction.

21. The processor of claim 20 wherein the memory execution unit includes an invalid store flag indicating if a store instruction has not execute properly and an invalid address flag indicating whether a store instruction executed improperly due to an invalid data or an invalid address.